Title: "GRAPHIC ENGINE FOR RASTERIZING A STRAIGHT EDGE IN A RESOURCE-CONSTRAINED DEVICE"
Inventor(s): Jiangen Cao
ARC-P120
1/12

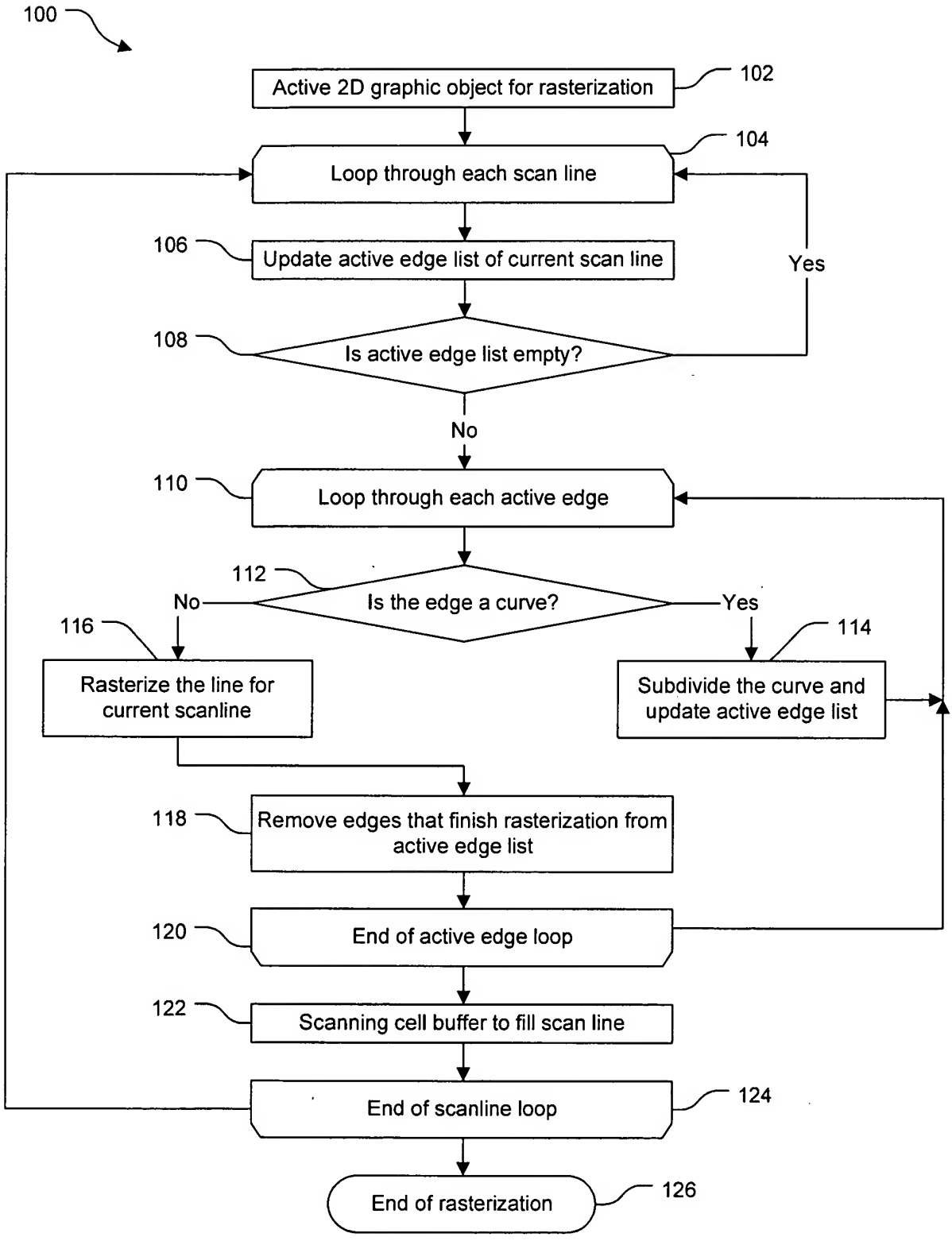

Fig. 1

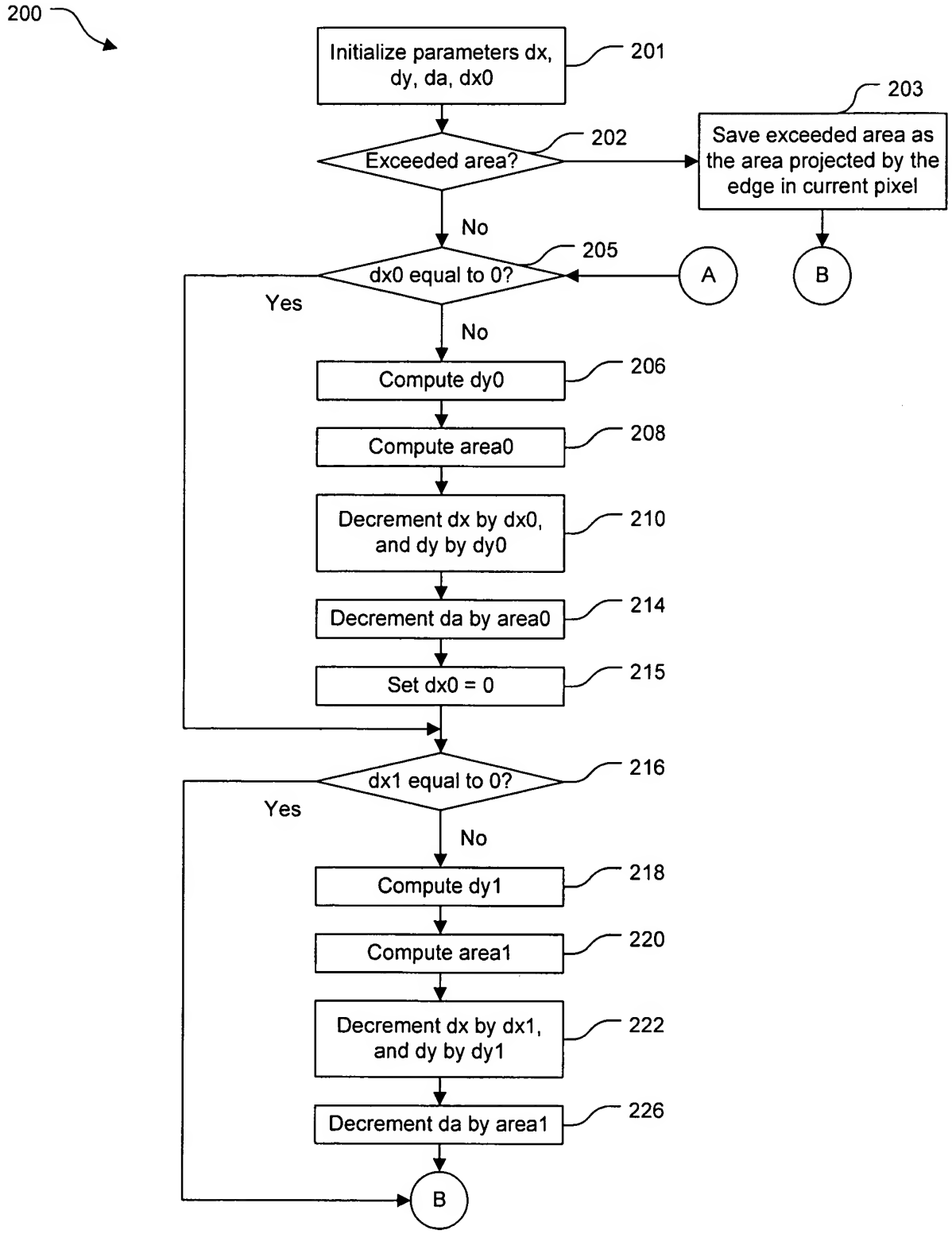
200
Initialize parameters dx, dy, da, dx0
201
Exceeded area?
202
Save exceeded area as the area projected by the edge in current pixel
203
No
dx0 equal to 0?
205
A
B
Yes
No
Compute dy0
206
Compute area0
208
Decrement dx by dx0, and dy by dy0
210
Decrement da by area0
214
Set dx0 = 0
215
dx1 equal to 0?
216
Yes
No
Compute dy1
218
Compute area1
220
Decrement dx by dx1, and dy by dy1
222
Decrement da by area1
226
B

Fig. 2A

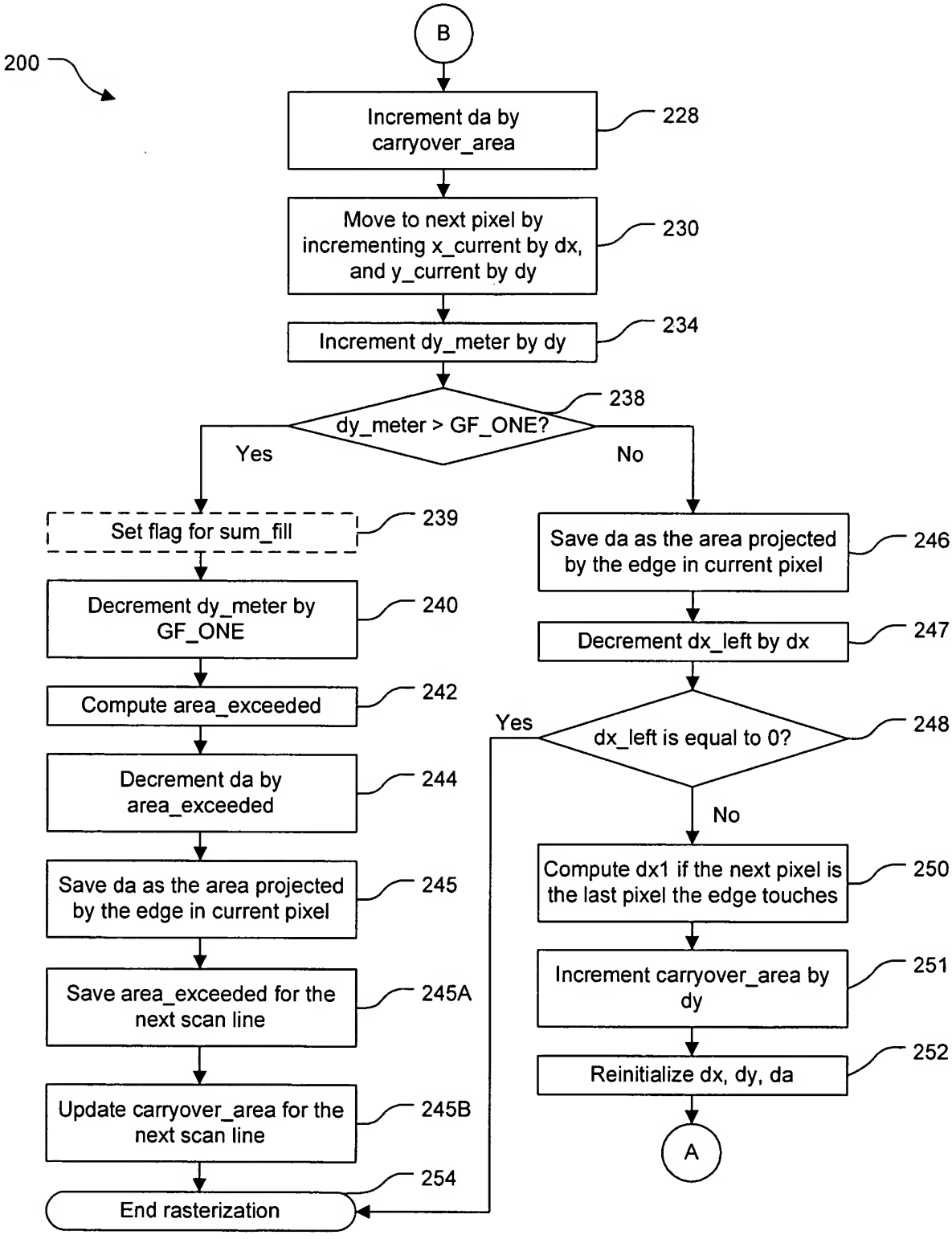
200
B
Increment da by carryover_area
228
Move to next pixel by incrementing x_current by dx, and y_current by dy
230
Increment dy_meter by dy
234
dy_meter > GF_ONE?
238
Yes
No
Set flag for sum_fill
239
Decrement dy_meter by GF_ONE
240
Compute area_exceeded
242
Decrement da by area_exceeded
244
Save da as the area projected by the edge in current pixel
245
Save area_exceeded for the next scan line
245A
Update carryover_area for the next scan line
245B
End rasterization
254
Save da as the area projected by the edge in current pixel
246
Decrement dx_left by dx
247
dx_left is equal to 0?
248
Yes
No
Compute dx1 if the next pixel is the last pixel the edge touches
250
Increment carryover_area by dy
251
Reinitialize dx, dy, da
252
A

Fig. 2B

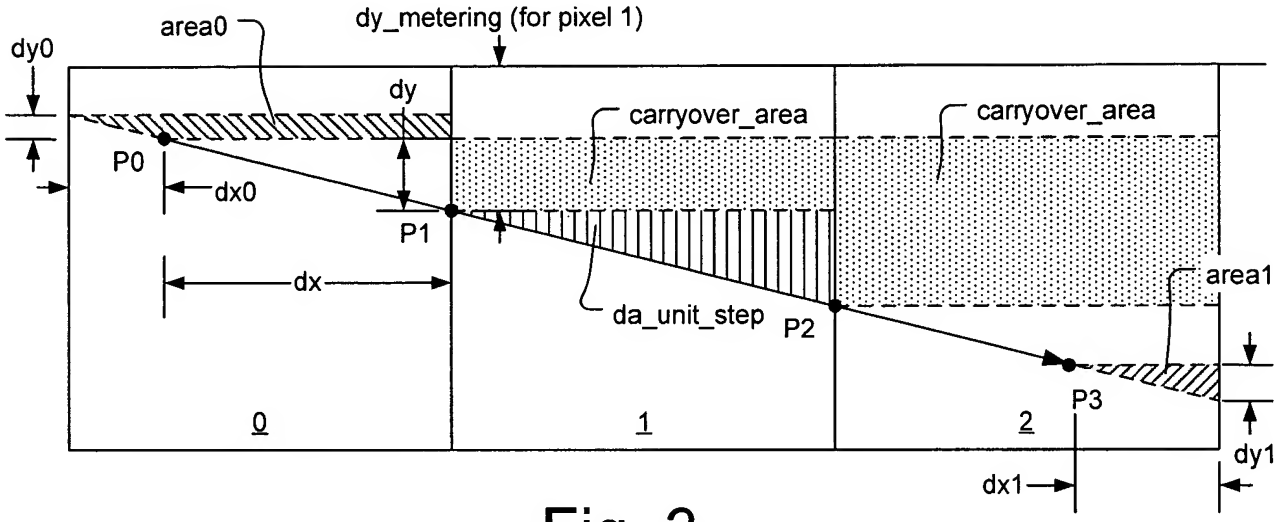
dy0
area0
dy_metering (for pixel 1)
dy
carryover_area
carryover_area
P0
dx0
P1
dx
da_unit_step
P2
area1
P3
0
1
2
dy1
dx1

Fig. 3

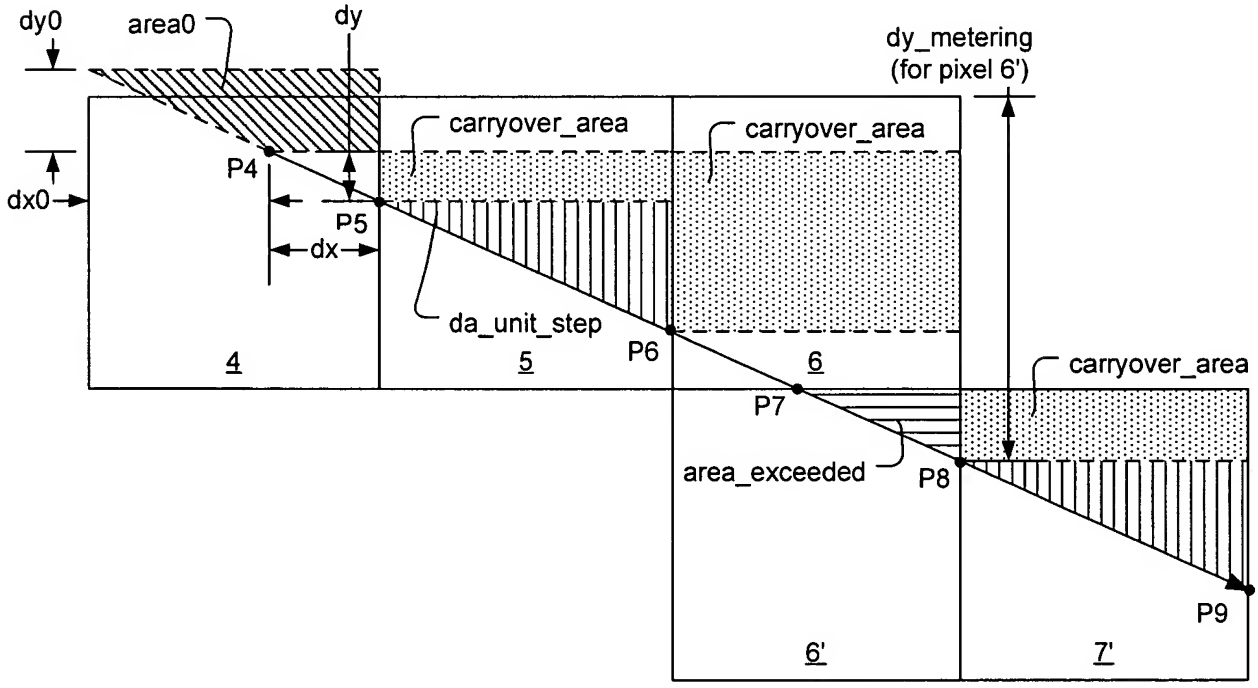
dy0
area0
dy
dy_metering
(for pixel 6')
carryover_area
carryover_area
P4
dx0
P5
dx
da_unit_step
P6
4
5
6
carryover_area
P7
area_exceeded
P8
P9
6'
7'

Fig. 4

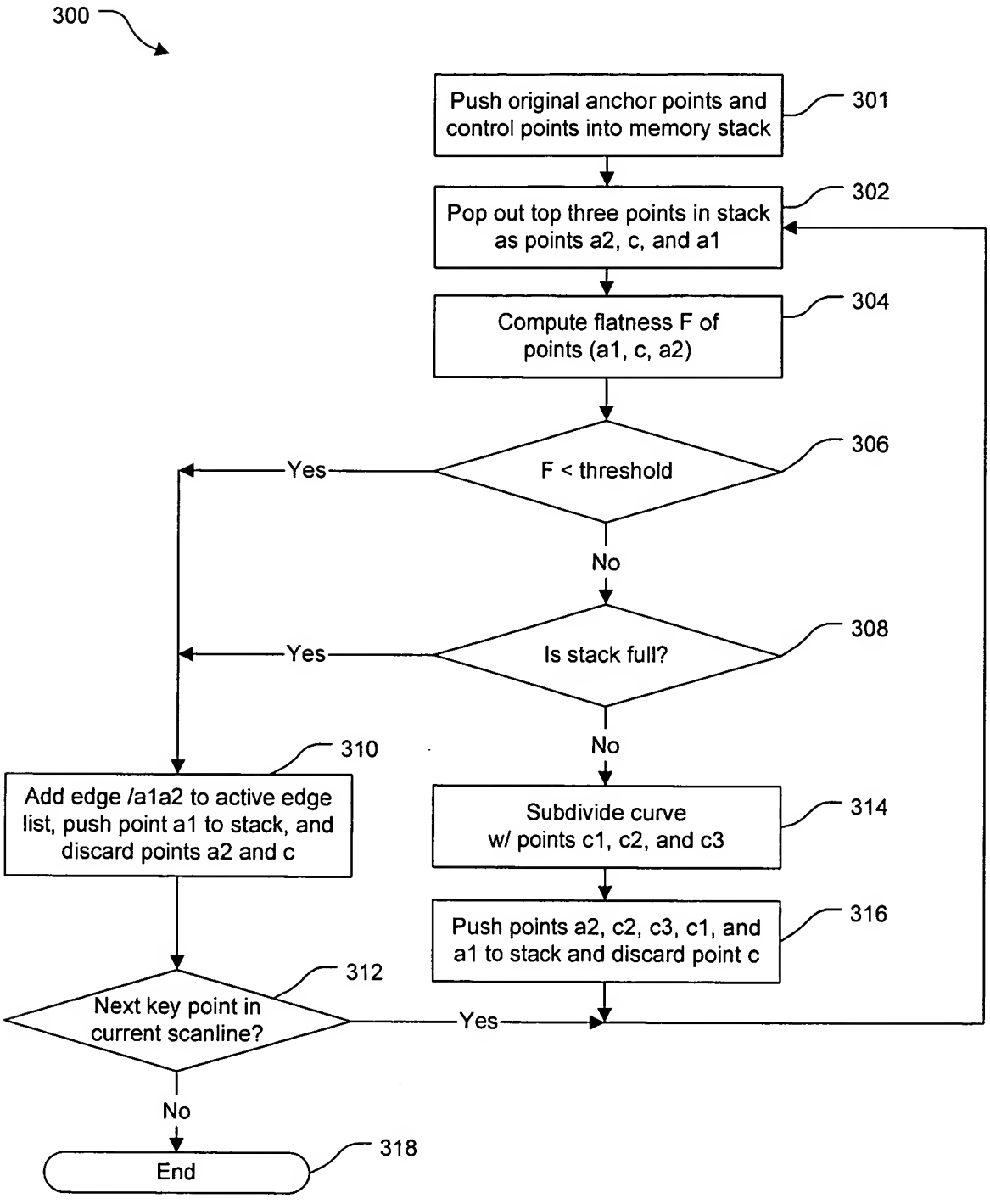
300
Push original anchor points and control points into memory stack
301
Pop out top three points in stack as points a2, c, and a1
302
Compute flatness F of points (a1, c, a2)
304
F < threshold
306
Yes
No
Is stack full?
308
Yes
No
Add edge /a1a2 to active edge list, push point a1 to stack, and discard points a2 and c
310
Subdivide curve w/ points c1, c2, and c3
314
Push points a2, c2, c3, c1, and a1 to stack and discard point c
316
Next key point in current scanline?
312
Yes
No
End
318

Fig. 5

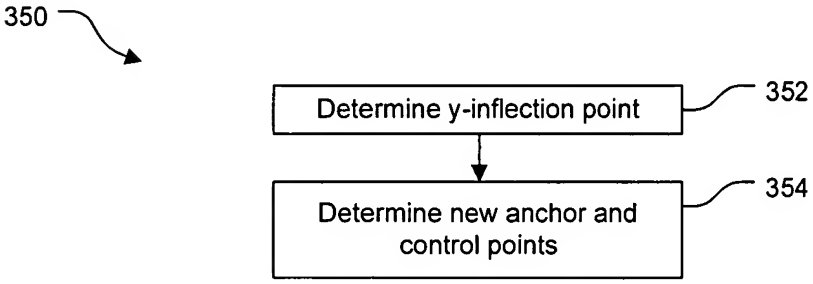
350
Determine y-inflection point
352
Determine new anchor and control points
354

Fig. 6

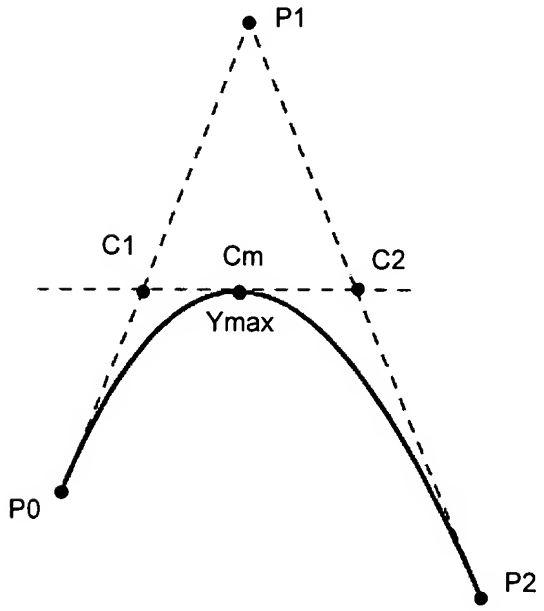
P1
C1
Cm
C2
Ymax
P0
P2

Fig. 6A

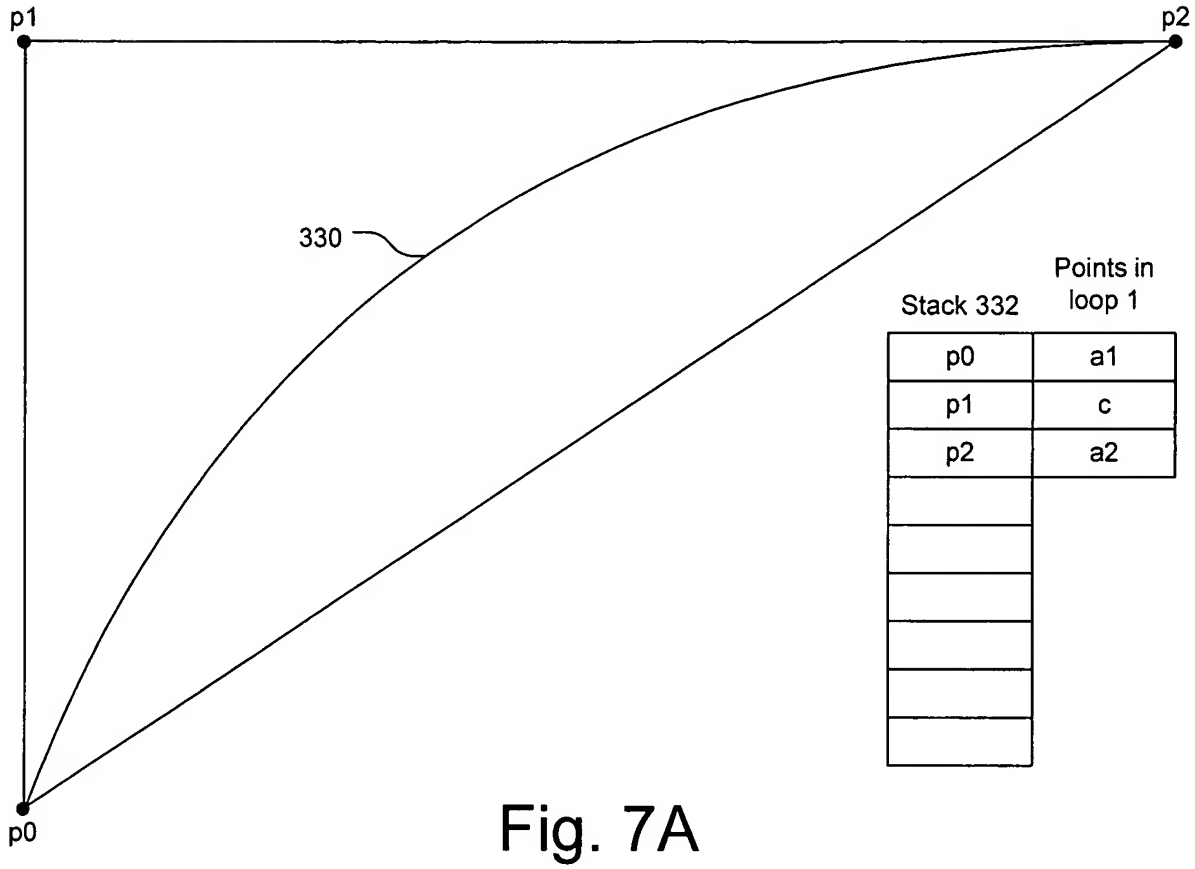
p1
p2
330
p0
Stack 332
Points in loop 1
p0
a1
p1
c
p2
a2

Fig. 7A

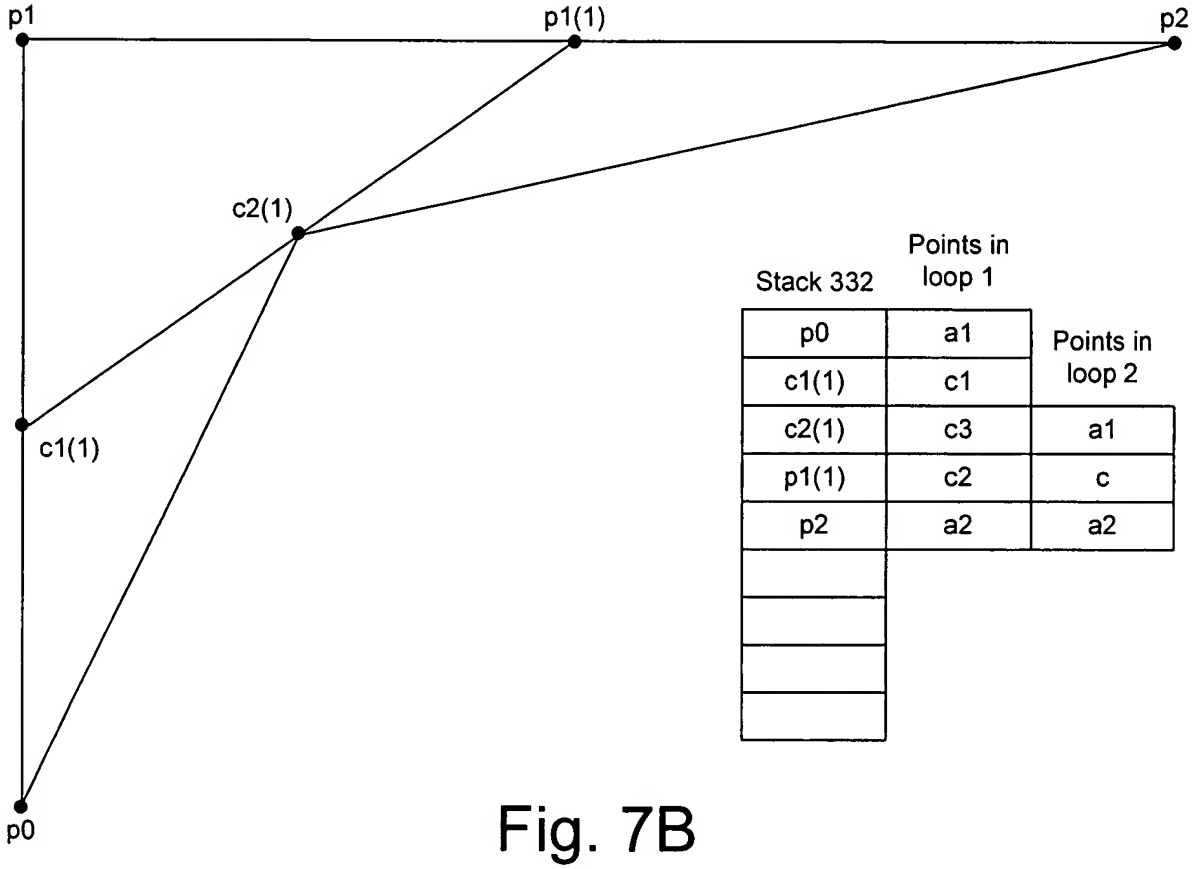
p1
p1(1)
p2
c2(1)
c1(1)
p0
Stack 332
Points in loop 1
Points in loop 2
p0
a1
c1(1)
c1
c2(1)
c3
a1
p1(1)
c2
c
p2
a2
a2

Fig. 7B

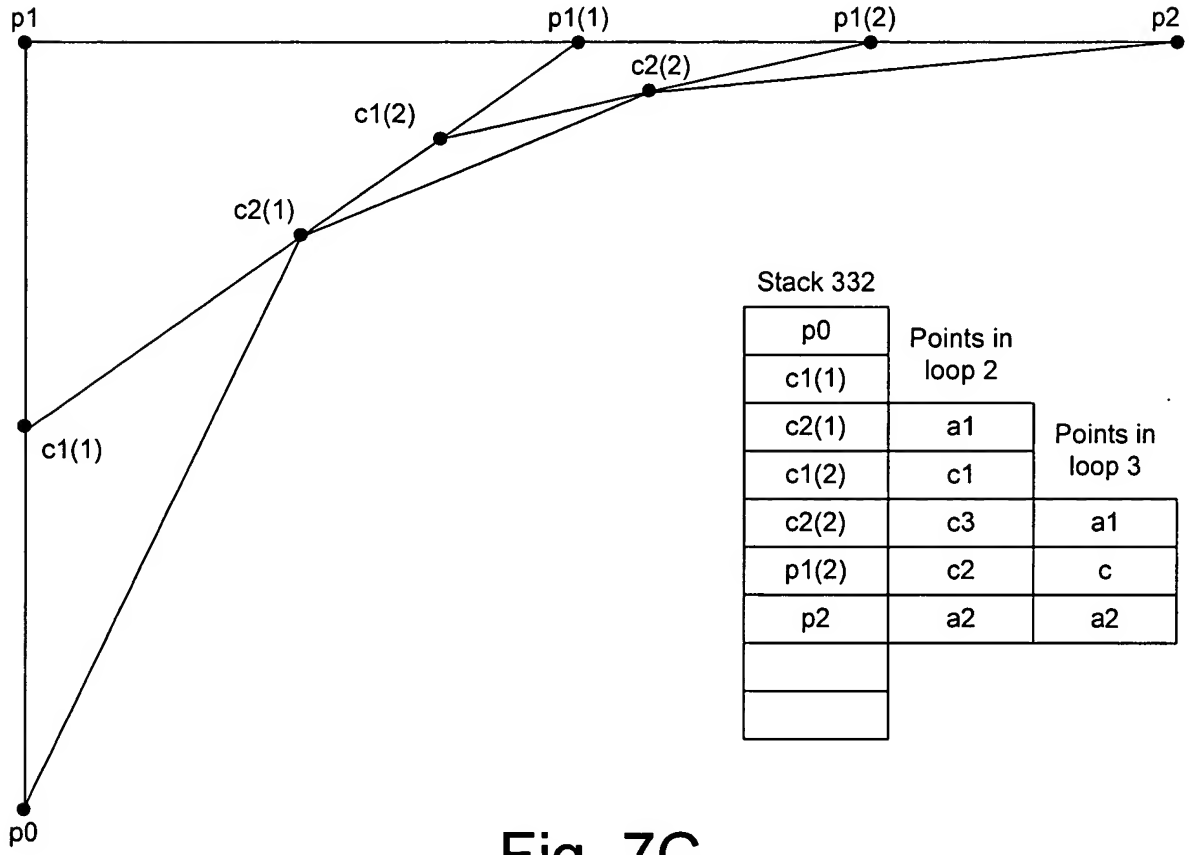
p1
p1(1)
p1(2)
p2
c2(2)
c1(2)
c2(1)
c1(1)
p0
Stack 332
p0
c1(1)
c2(1)
c1(2)
c2(2)
p1(2)
p2
Points in loop 2
a1
c1
c3
c2
a2
Points in loop 3
a1
c
a2

Fig. 7C

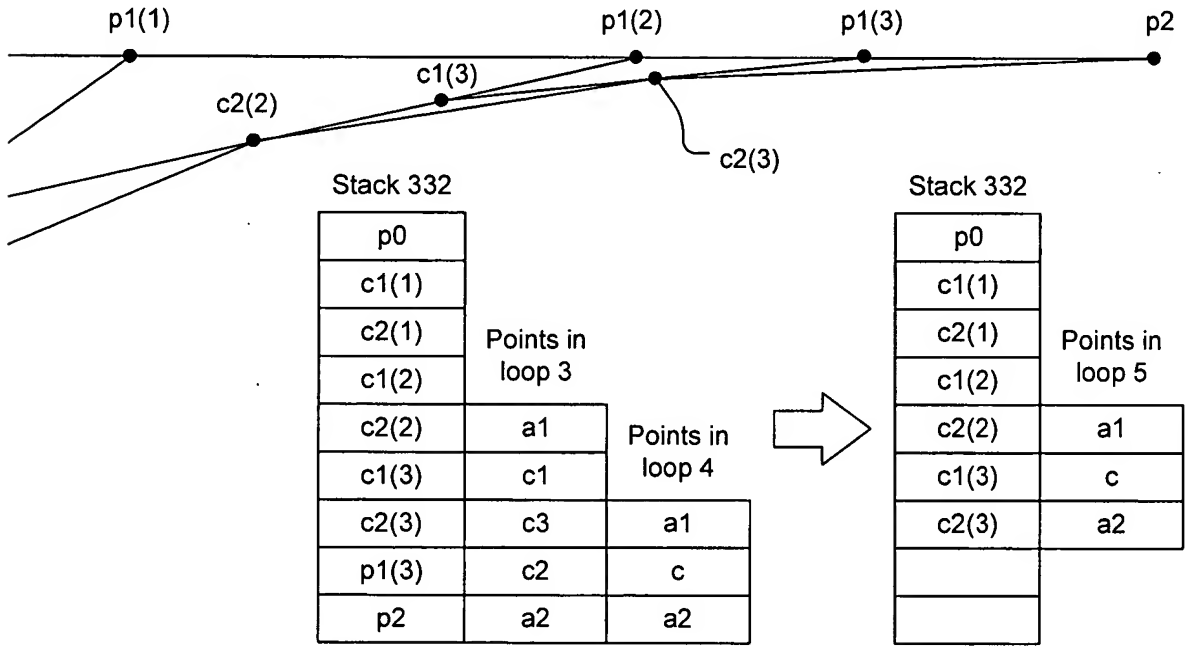
p1(1)
p1(2)
p1(3)
p2
c1(3)
c2(2)
c2(3)
Stack 332
p0
c1(1)
c2(1)
c1(2)
c2(2)
c1(3)
c2(3)
p1(3)
p2
Points in loop 3
a1
c1
c3
c2
a2
Points in loop 4
a1
c
a2
Stack 332
p0
c1(1)
c2(1)
c1(2)
c2(2)
c1(3)
c2(3)
Points in loop 5
a1
c
a2

Fig. 7D

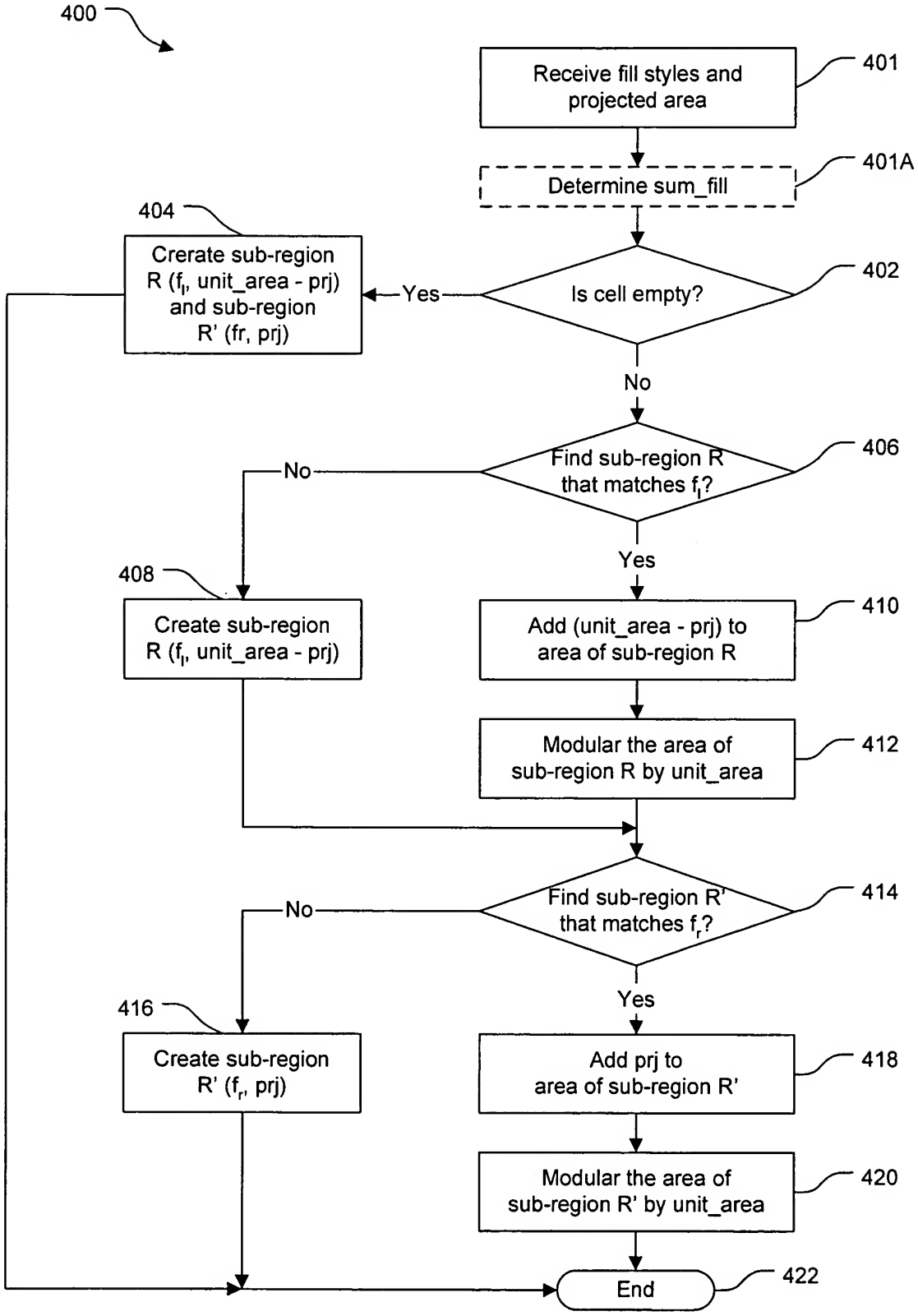
400
Receive fill styles and projected area
401
Determine sum_fill
401A
Is cell empty?
402
Yes
Crerate sub-region R (f_l, unit_area - prj) and sub-region R' (fr, prj)
404
No
Find sub-region R that matches f_l?
406
No
Yes
Create sub-region R (f_l, unit_area - prj)
408
Add (unit_area - prj) to area of sub-region R
410
Modular the area of sub-region R by unit_area
412
Find sub-region R' that matches f_r?
414
No
Yes
Create sub-region R' (f_r, prj)
416
Add prj to area of sub-region R'
418
Modular the area of sub-region R' by unit_area
420
End
422

Fig. 8

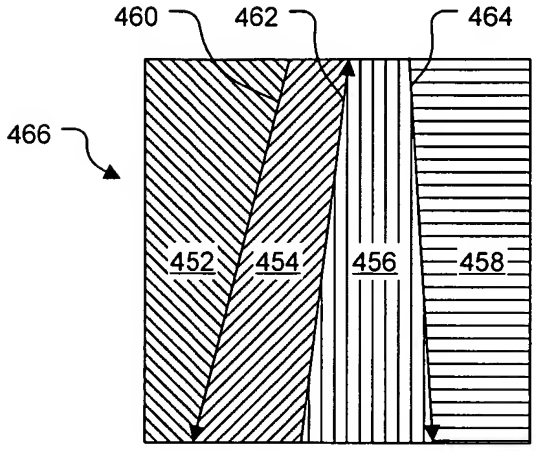

Fig. 9A

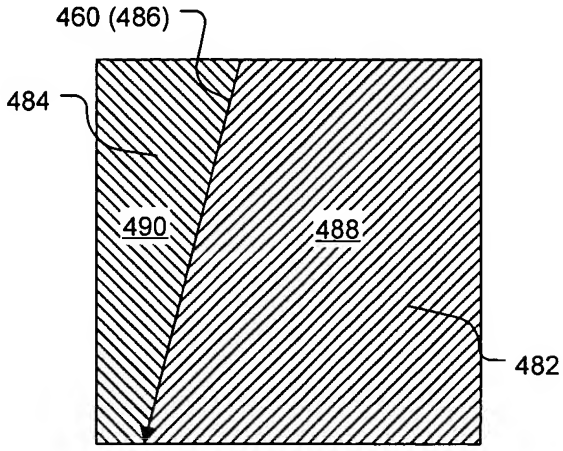

Fig. 9B

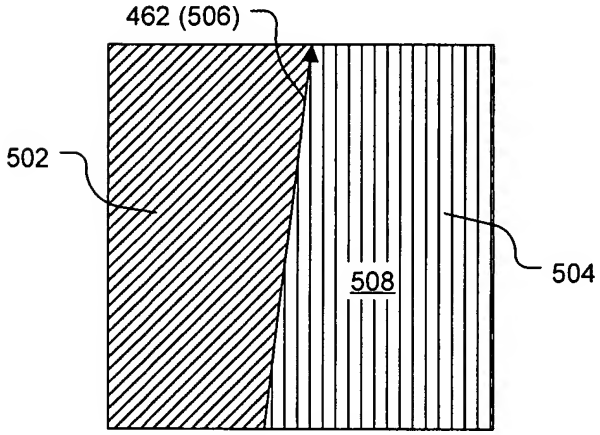

Fig. 9C

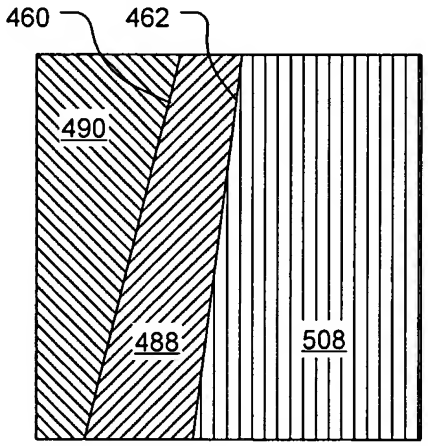

Fig. 9D

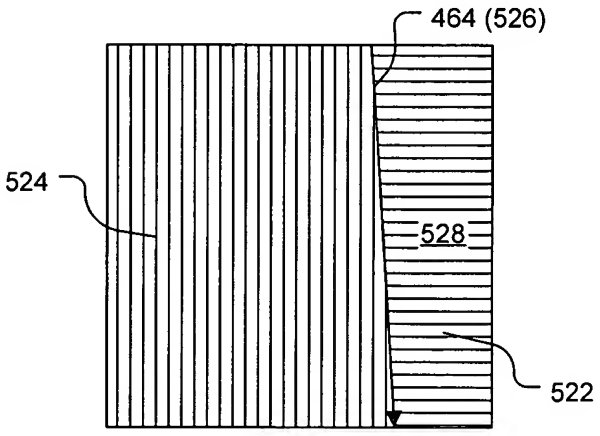

Fig. 9E

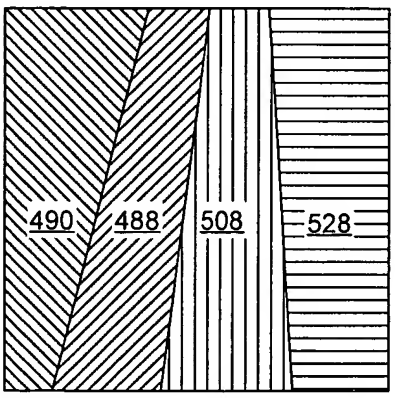

Fig. 9F

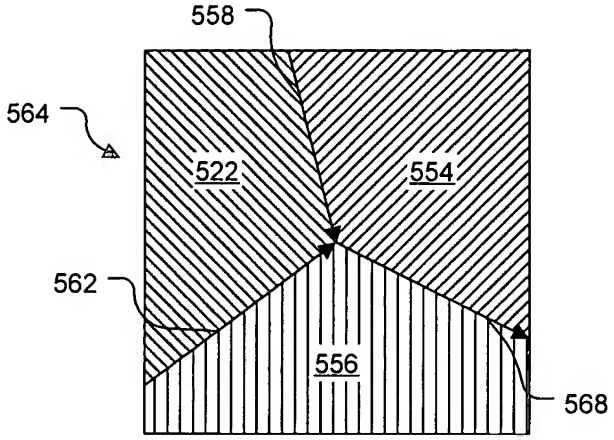

Fig. 10A

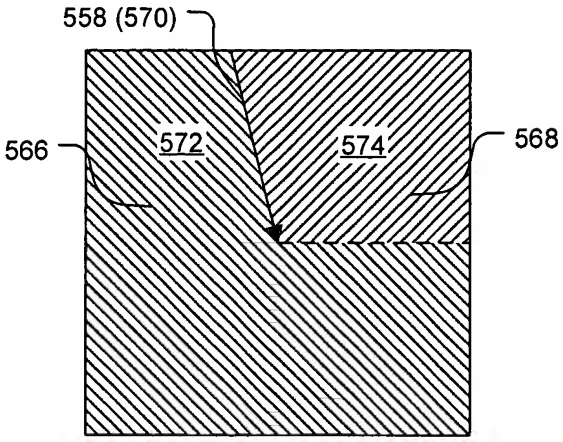

Fig. 10B

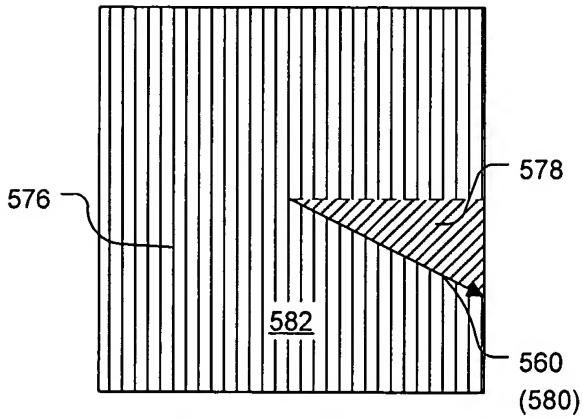

Fig. 10C

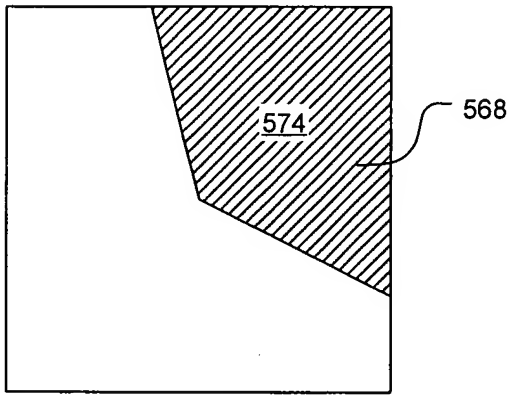

Fig. 10D

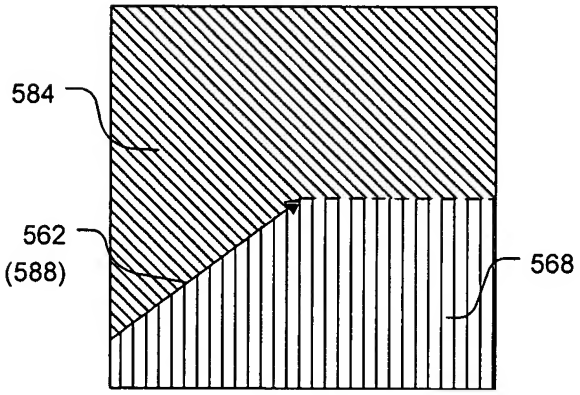

Fig. 10E

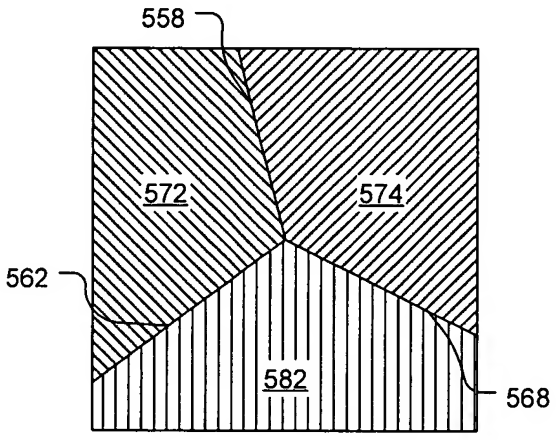

Fig. 10F

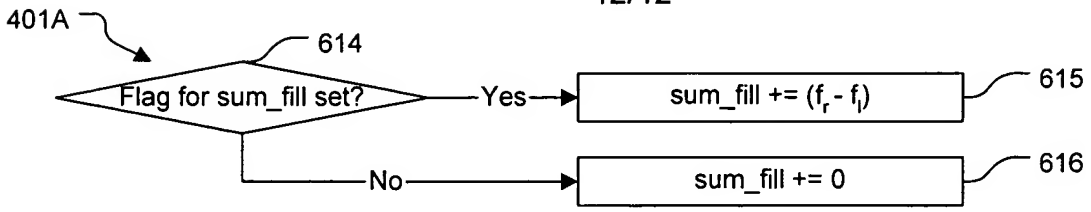

Fig. 11A

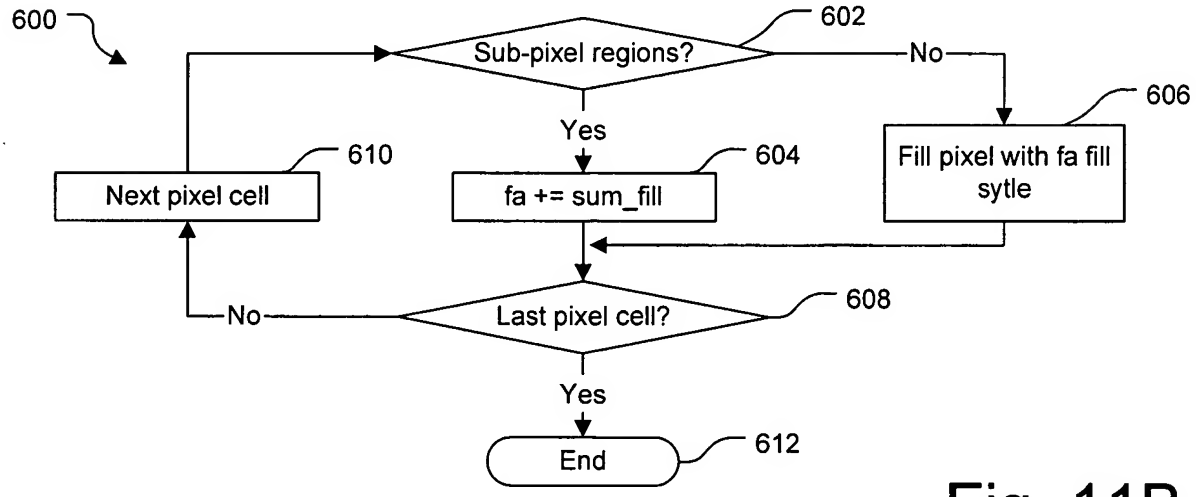

Fig. 11B

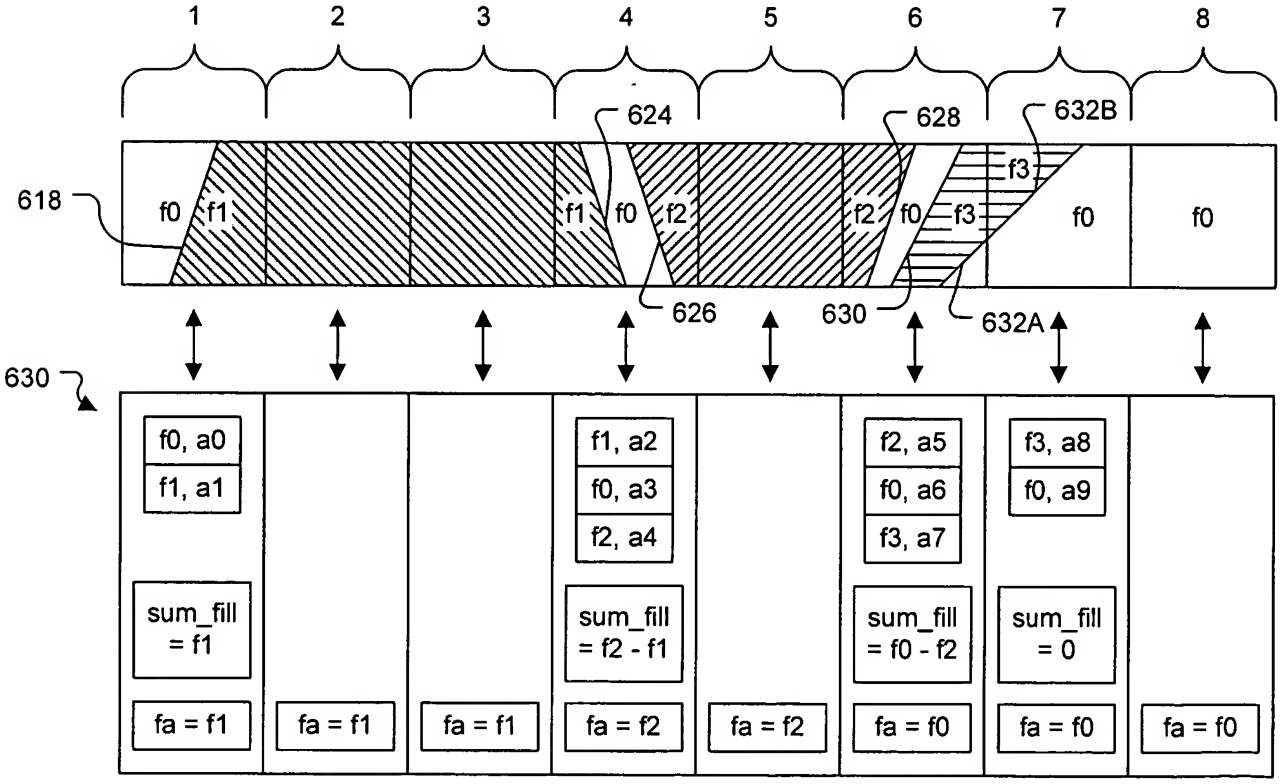

Fig. 12